

CS10003 Programming and Data Structures

Spring Semester, 2021-2022

Sections 7 & 8

Long Test 1 – Part B[Marks = 50]

18-May-2022, 10:15 to 11:25

INSTRUCTIONS

- You have 1 hour for writing and 10 minutes for submission.
 - Write your answers on paper. Answers must be handwritten. Typed or written answers using an electronic device are not allowed.
 - Write your name and roll number on each page. Write page numbers for each page.
 - Scan all pages and collate. Create a single PDF file (of size ≤ 10 MB). You could also take pictures of different pages, combine them to make a single pdf file.
 - The name of the file for this part should be `<RollNumber>.Long1B`. Ensure your roll number in the filename has only digits and uppercase characters.
 - Upload your file. Make sure you click on the 'Turn in' button to submit your file.
 - The said deadlines are strict after which no submissions will be allowed.
 - We will not accept submission by any other means.
-

1. Write a program that prints the Fibonacci *words* of order 0 through 10. Let $f(i)$ denote the Fibonacci word of order i . We have $f(0) = "a"$, $f(1) = "b"$, $f(2) = "ba"$, $f(3) = "bab"$, $f(4) = "babba"$, and in general $f(n) = f(n-1)$ followed by $f(n-2)$. Use string concatenation. There should be only a `main()` function in your program. 10

Solution:

```
#include <stdio.h>
#include <string.h>

int main(){
    char f0[100] = "a", f1[100] = "b", f2[100];
    int i,j,len0,len1;
    printf(" 0: %s\n 1: %s\n",f0,f1);
    for(i=2; i<=10; i++){
        strcpy(f2,f1);
        len0 = strlen(f0);
        len1 = strlen(f1);
        for(j=len1; j-len1 <= len0; j++)
            f2[j] = f0[j-len1];
        printf("%2d: %s\n",i,f2);
        strcpy(f0,f1);
        strcpy(f1,f2);
    }
    return 0;
}
```

2. A word s is called x -ish if every letter of x is contained in s in some position. Write a program that, given two words s and x (read as strings), detects if s is x -ish or not. For example, if $x = \text{"elf"}$, then the word "tasteful" is x -ish (or elf-ish) whereas the word "malfunction" is not elf-ish. Assume that the two input strings will have lengths at most 49. Also, assume that the input strings contain only lowercase letters. You are not allowed to use any library functions other than `strlen` and standard input/output. Your program should have only the `main()` function. 10

Solution:

```
#include <stdio.h>
#include <string.h>

int main(){
    char s[50], x[50];

    int i, j;
    int lens, lenx;
    printf("Enter s: ");
    scanf("%s", s);
    printf("Enter x: ");
    scanf("%s", x);

    int flag = 0;
    lens = strlen(s);
    lenx = strlen(x);
    for(i=0; i<lenx; i++){
        flag = 0;
        for(j=0; j<lens; j++){
            if(x[i] == s[j])
                flag = 1;
        }
        if(flag == 0){
            printf("%s is not %s-ish.\n",s,x);
            return 0;
        }
    }
    printf("%s is %s-ish.\n",s,x);
    return 0;
}
```

3. Write a C function `void pascal(int n)` that prints the following pattern for $n = 5$.

```
0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 0
0 1 2 2 2 2 2 1 0
0 1 2 3 3 3 2 1 0
0 1 2 3 4 3 2 1 0
0 1 2 3 3 3 2 1 0
0 1 2 2 2 2 2 1 0
0 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0
```

An incomplete program is given below. Fill in the blanks so that the function works for any number n .

```
#include<stdio.h>

#define minimum(a, b) _____ (A)
```

```

void pascal(int n) {
    int row, col, min;

    /* len denotes the number of rows or columns in terms of n */
    int len = _____; (B)
    for(row = 0; row < len; row++) {
        for(col = 0; col < len; col++) {
            min = minimum(_____, _____); (C)
            min = minimum(min, _____); (D)
            min = minimum(min, _____); (E)
            printf("%d ", min);
        }
        printf("\n");
    }
    return;
}

```

10

Solution:

- (A) (a) < (b) ? (a) : (b)
- (B) n * 2 - 1
- (C) row, col
- (D) len - row - 1
- (E) len - col - 1

4. We need to sort (in non-decreasing order), an input unsorted array A of n integers where n is a power of 2. First we compare $n/2$ mutually disjoint (consecutive pairs), and set each such pair in right order. If $n = 8$ and $A = \{8, 7, 6, 5, 4, 3, 2, 1\}$ was the input then we have $A = \{7, 8, 5, 6, 3, 4, 1, 2\}$ after the first iteration. We will have exactly k iterations where $n = 2^k$. In iteration 2, A will become $A = \{5, 6, 7, 8, 1, 2, 3, 4\}$ because we will process $4 = n/2 = 8/2$ pairs of sequences of length two – e.g., $\{7, 8\}$ and $\{5, 6\}$, to get $\{5, 6, 7, 8\}$ and $\{3, 4\}$ and $\{1, 2\}$ to get $\{1, 2, 3, 4\}$ resulting in $A = \{5, 6, 7, 8, 1, 2, 3, 4\}$. Finally, in the third iteration, we will merge only one pair of length four sequences $\{5, 6, 7, 8\}$ and $\{1, 2, 3, 4\}$ to get the sorted $A = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Write an iterative C program with arrays but no recursive function for doing this job. Print the inputs n and A and the sorted array A . Assume that $n \leq 2^{10}$.

10

Solution:

```

#include<stdio.h>
#include<math.h>

void merge ( int a[ ], int lo, int m, int hi ) {
    int i, j, k;
    int b[1024];

    for (i=lo; i<=hi; i++)
        b[i]=a[i];

    i=lo; j=m; k=lo;
    while (i<m && j<=hi)
        if (b[i]<=b[j]) a[k++]=b[i++];
        else a[k++]=b[j++];
}

```

```

    while (i<m) a[k++]=b[i++];
    while (j<=hi) a[k++]=b[j++];
}

int main() {

    int n,i,j,a[1024],k;
    int noloops,lo,mid,hi,logn,nn;

    scanf("%d",&n);
    for (i=0;i<n;i++)
        scanf("%d",&a[i]);

    printf("\nInput array\n");
    for (i=0;i<n;i++)
        printf("%d ",a[i]);
    printf("\n");

    nn=n; logn=0;
    while (nn>1){
        ++logn;
        nn=nn/2;
    }
    int k = 1;
    for (i=0;i<logn;i++) {
        noloops=(n/(2*(i+1)));
        for (j=0;j<noloops;j++) {
            lo=j*2*k;
            mid=lo+k;
            hi=mid+k-1;
            merge(a,lo,mid,hi);
        }
        k *= 2;
    }

    printf("\nSorted array\n");
    for (i=0;i<n;i++)
        printf("%d ",a[i]);
    printf("\n");
}

```

5. Write a recursive function that takes a positive integer n as its argument and returns the largest digit in n as the result. For example, if $n = 51398$, the function should return 9. The function should not use any array, nor should it have any loop. Write both the function and the main program. Read the input in the main program and print the input read and result (obtained on calling the recursive function) in the main program.

10

Solution:

```

#include<stdio.h>

int LarDig(int n) {
    int max, val;
    if (n < 10) return n;
    max = n%10;
    val = LarDig(n/10);
    if (max >= val) return max;
    else return val;
}

```

```
main() {  
    int numb, result;  
    scanf("%d", &numb);  
    printf("numb = %d, LarDig = %d \n", numb, LarDig(numb));  
}
```